SQLite

TW. Web Technologies Course MM, 2020/21 Edition

Sérgio Nunes DEI, FEUP, U.Porto

Database Management Systems

- → Database Management Systems (DBMS) are software applications that provide features do define, create, maintain, and control databases.
- → DBMS are typically targeted at a specific database model, such as the relational model (RDBMS), the object oriented (OODBMS), or others.
- → DBMS architectures typically follow a client-server model, where clients (e.g. applications) interact with a centralized server managing the database.
- → An alternative architecture is a serverless model where access to the database is done via library calls, without message passing between different processes. The most popular example is SQLite.
- → Question: Where can we see databases in use?

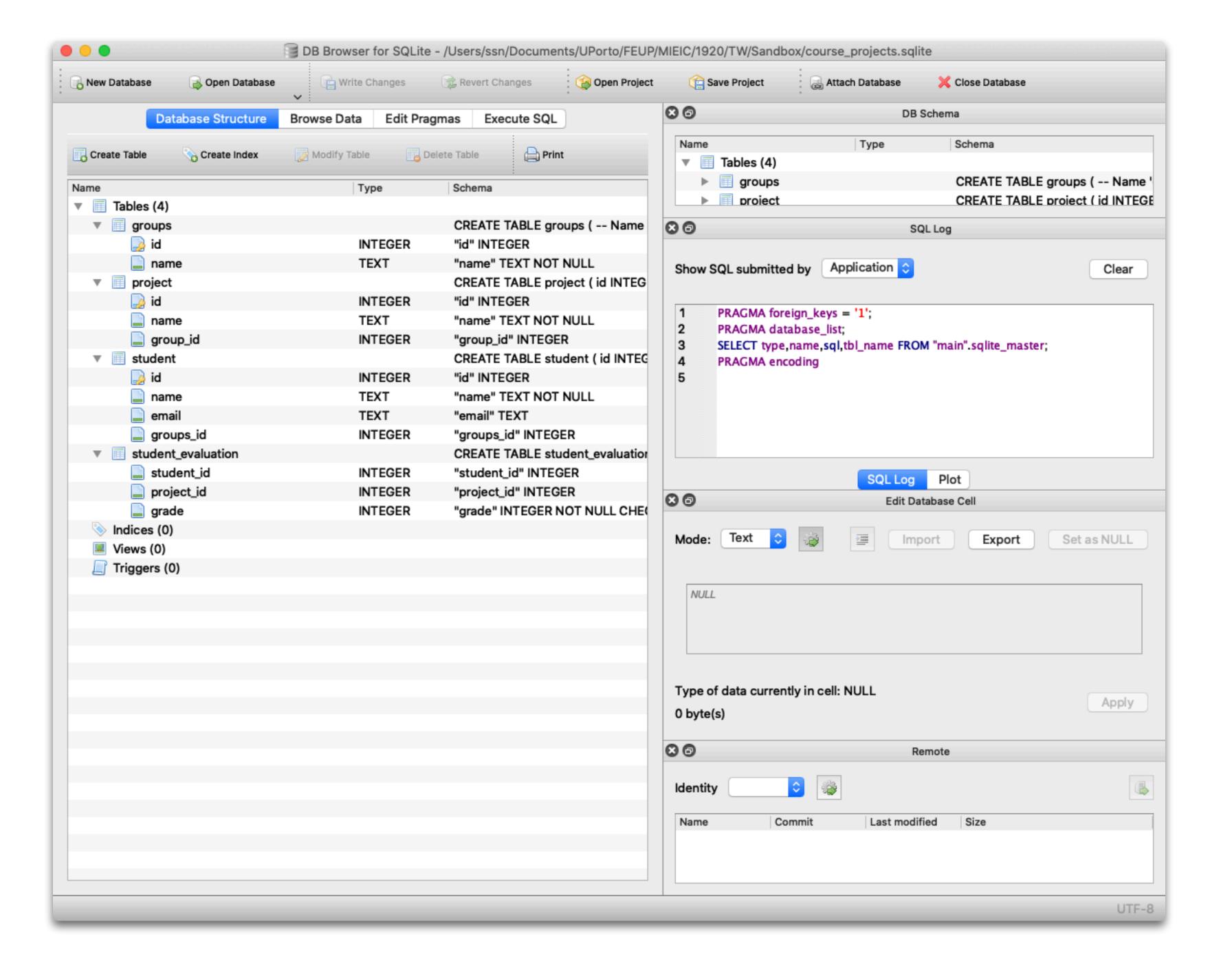
DBMS Features

- → Features that a DBMS can provide:
 - → Data storage, retrieval and update;
 - → Constrain enforcement over data;
 - → Schema management;
 - → Support for transactions and concurrency;
 - → Recovery from failure;
 - → Access control for data interaction;
 - → And many more...
- → Popular commercial RDBMS include: Oracle, MS SQL Server, IBM DB2.
- → Open-source RDBMS include: PostgreSQL, MySQL, MariaDB, SQLite.

SQLite

- → SQLite is a serverless, ACID compliant, open-source RDBMS.
- → It is commonly used in mobile phones, operating systems, applications (e.g. browsers), amount others. It is the most used database in the world.
- → A SQLite database is a single file.
- → Interaction with a SQLite database is done via library calls.

→ http://sqlite.org



Install SQLite

- → SQLite is provided as an executable file.
- → Simply download your operating system version and copy the sqlite3 / sqlite3.exe to your project folder.
- → If you plan to use SQLite in other projects you should copy SQLite to a folder that is accessible system-wide. This requires some additional knowledge on configuring the system path.

→ If you open the sqlite3 application (e.g. double click) you will see the SQLite command line interface. You can exit this interface by entering the command ".quit".

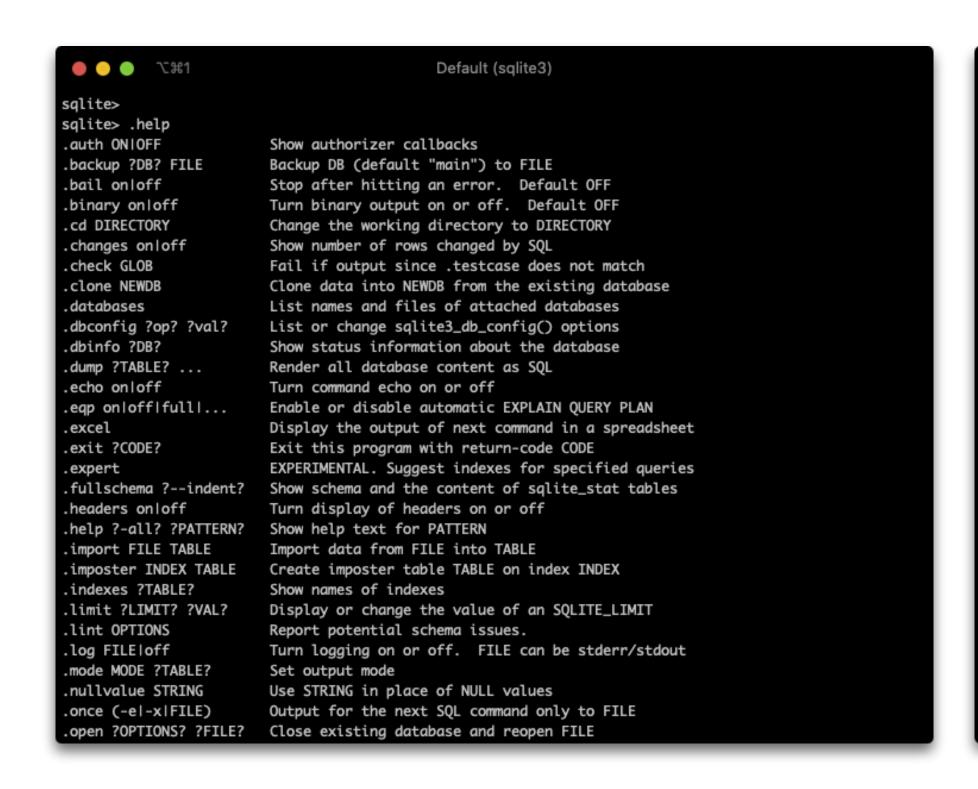
Ongoing example

- → Download an example SQLite database from:
 - → https://drive.google.com/file/d/10mc8TaqNrvwdlv_tlZTXfHoeaMXlUyFX/view?usp=sharing

- → Copy this file to the project's folder, where sqlite3 is.
- → Open sqlite3 and then open the database with the command:
 - → .open course_projects.sqlite

Using SQLite

→ You can type ".help" in the SQLite command line interface to see a summary of all the available commands.



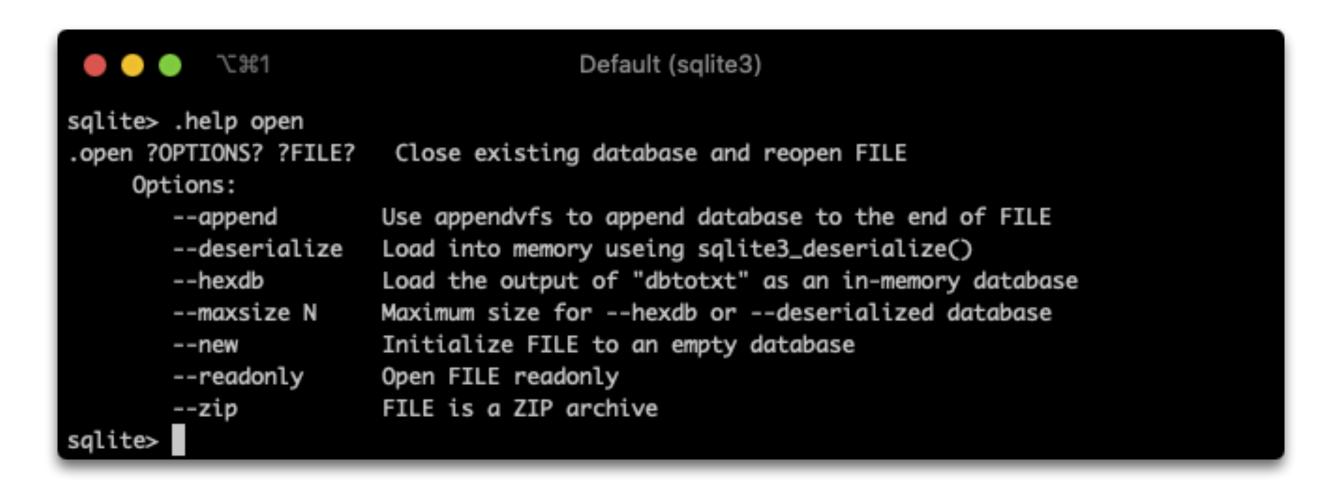


SQLite Commands

- → SQLite commands can be divided in three types:
 - → Meta Commands, which are used to interact with databases (e.g. open, view, save), view or define SQLite settings, and other administrative operations. These commands start with a dot (.) .help
 - → SQL Data Definition Language (DDL), standard SQL commands to create, alter, and delete database tables.
 - → SQL Data Manipulation Language (DML), standard SQL commands to query, add, alter, and delete data.

.help

- → .help ?-all? ?PATTERN?
- → Show help text for PATTERN.
- → [.help] shows a list of available commands.
- → [.help <command>] provides details about a specific command.



.open

- → .open ?OPTIONS? ?FILE?
- → Close existing database and reopen FILE.
- → [.open --new] creates a new empty database.
- → [.open --readonly] opens FILE in readonly mode.

```
Default (sqlite3)

sqlite>
sqlite>
sqlite>
sqlite>
sqlite> .open course_projects.sqlite
sqlite> .tables
groups project student student_evaluation
sqlite>
```

.tables

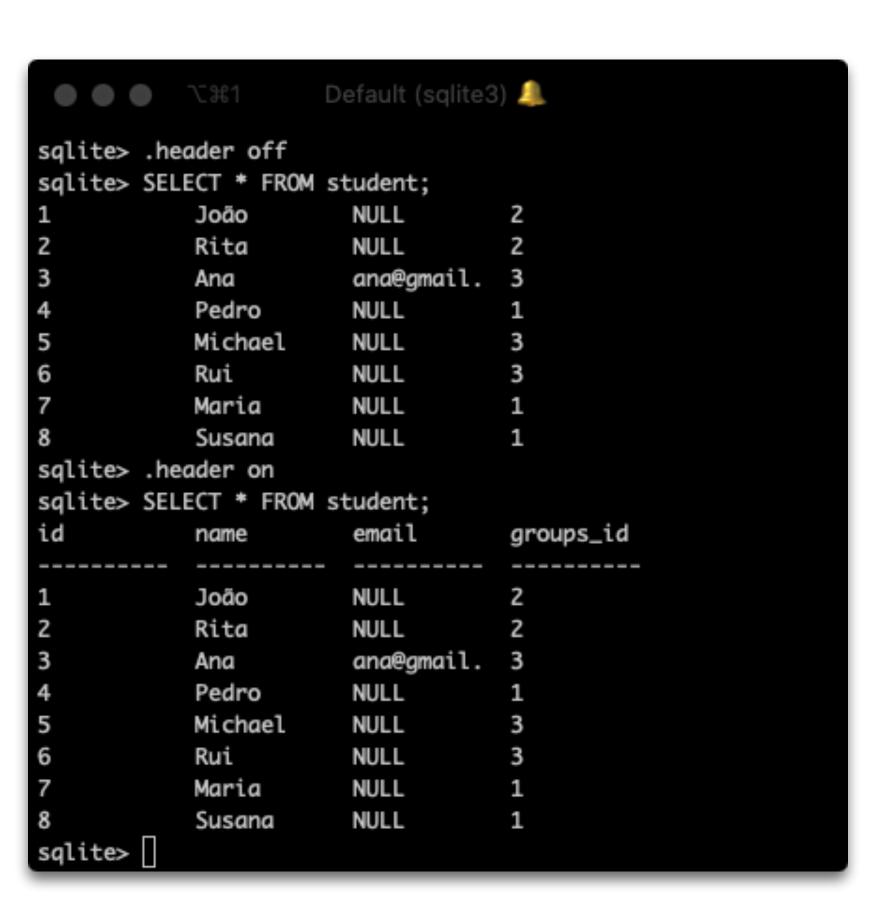
- → .tables ?TABLE?
- → List names of tables matching LIKE pattern TABLE.
- → [.tables] lists all tables in the currently open database.

.schema

- → .schema ?--indent? ?PATTERN?
- → Show the CREATE statements matching PATTERN.
- → The indent option tries to pretty-print the schema.

.header

- → .headers on off
- → Turn display of headers on or off.



.mode

- → .mode MODE ?TABLE?
- → Set output mode (e.g. csv, column, html, insert, line, etc).
- → [.mode column] shows results using the default table layout.

```
sqlite> sqlite> .mode csv
sqlite> SELECT * FROM student;
id,name,email,groups_id
1,"João",NULL,2
2,Rita,NULL,2
3,Ana,ana@gmail.com,3
4,Pedro,NULL,1
5,Michael,NULL,3
6,Rui,NULL,3
7,Maria,NULL,1
8,Susana,NULL,1
sqlite>
sqlite>
```

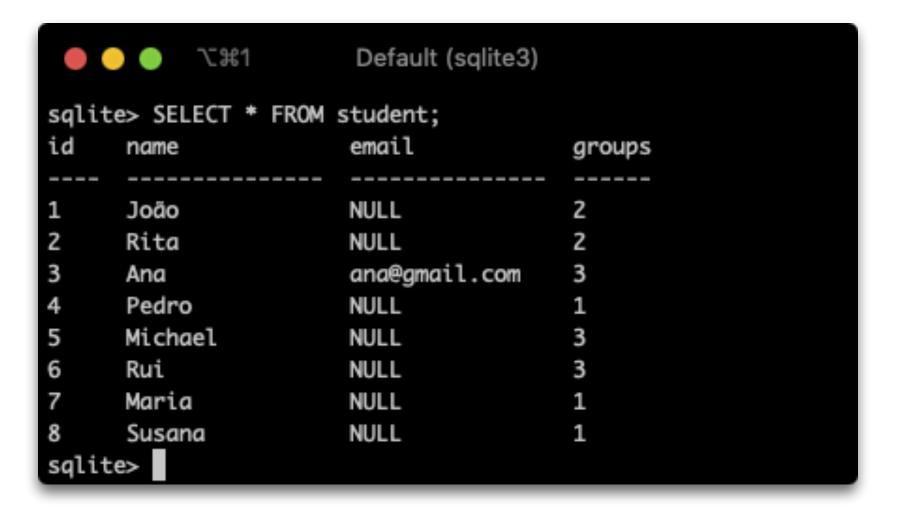
```
1第7 🔵 🔵
                        Default (sqlite3)
sqlite> .mode column
sqlite> SELECT * FROM student;
                        email
                                   groups_id
            João
                       NULL
                       NULL
            Rita
                       ana@gmail. 3
            Ana
                       NULL
            Pedro
                       NULL
            Michael
                       NULL
            Rui
                       NULL
            Maria
            Susana
                       NULL
sqlite>
```

.width

```
→ .width NUM1 NUM2 ...
```

→ Set column width (in characters) for column mode.

```
● ● ● ₹3
                      Default (sqlite3)
sqlite> SELECT * FROM student;
                           email
           name
                                           groups
           João
                           NULL
           Rita
                           NULL
                           ana@gmail.com
           Ana
                           NULL
           Pedro
           Michael
                           NULL
                           NULL
           Rui
           Maria
                           NULL
           Susana
                           NULL
sqlite>
```



.read

- → .read FILE
- → Read input from FILE.
- → [.read mydb.sql] executes all SQL statements in the mysb.sql file.

.save

- → .save FILE
- → Write the in-memory database into FILE.
- → [.save mydb.sqlite] saves the currently active (in-memory) database to the mydb.sqlite file.

Install DB Browser for SQLite

- → DB Browser for SQLite is an open-source graphical user interface (GUI) to interact with SQLite databases.
- → https://sqlitebrowser.org
- → Download and install latest release (menu option "GitHub").
- → DB Browser can be used to open and alter existing databases or to create new SQLite databases.

Data Manipulation Language

- → Standard SQL can be used to query a database.
- → SELECT * FROM students;
- → SELECT student.name, groups.name
 FROM student, groups
 WHERE student.groups_id = groups.id;

```
sqlite>
sqlite>
sqlite>
sqlite> SELECT * FROM student;
                    email
     name
                                   groups
                    NULL
     João
     Rita
                    NULL
                    ana@qmail.com
     Ana
     Pedro
                    NULL
                    NULL
     Michael
                                   3
     Rui
                    NULL
                                   3
                    NULL
                                   1
     Maria
     Susana
                    NULL
                                   1
sqlite>
```

```
Sqlite>
sqlite>
sqlite> SELECT student.name, groups.name
...> FROM student, groups
...> WHERE student.groups_id = groups.id;
name name
...
João Grupo 2
Rita Grupo 2
Ana Grupo 3
Pedr Grupo 1
Mich Grupo 3
Rui Grupo 3
Mari Grupo 1
Susa Grupo 1
sqlite>
```

Data Definition Language

→ Standard SQL can be used to create, alter, or delete tables.

```
→ CREATE TABLE book (
   id INTEGER PRIMARY KEY,
   name TEXT
);

→ ALTER TABLE book RENAME TO books;

→ ALTER TABLE books
   ADD COLUMN author TEXT;

→ DROP TABLE books;
```

Creating a Database

- → To create a new database you create a text file (script) with all the SQL statements necessary to create the tables.
- → Then, you have two ways to create the SQLite database:
 - → In SQLite, with the command [.read script_file.sql]
 - → In the command line:
 - → Linux / OSX: [sqlite3 database.sqlite3 < script_file.sql]
 - → Windows: [sqlite3.exe DB.db ".read script_file.sql"]

Populating a Database

- → To insert data to a SQLite database, you create a text file with the SQL statements necessary to insert each record to each table.
- → You have two ways to execute this SQL script:
 - → In SQLite, with the command [.read script_inserts.sql]
 - → In the command line:
 - → Linux / OSX: [sqlite3 database.sqlite3 < script_inserts.sql]
 - → Windows: [sqlite3.exe DB.db ".read script_inserts.sql"]